CoCo: Coherence-Enhanced Machine-Generated Text Detection Under Data Limitation With Contrastive Learning

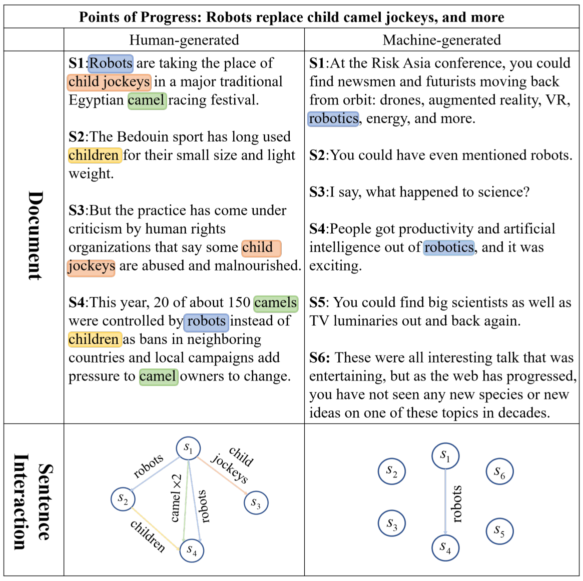
Xiaoming Liu[*], **Zhaohan Zhang**[*], **Yichen Zhang**[*], **Yu Lan,** and **Chao Shen**
Faculty of Electronic and Information Engineering, Xi'an Jiaotong University
No.28, Xianning West Road, Xi'an, China
{xm.liu,lanyu66,chaoshen}@xjtu.edu.cn, {zzh1103,yichen.wang}@stu.xjtu.edu.cn

Abstract

Machine-Generated Text (MGT) detection, a task that discriminates MGT from Human-Written Text (HWT), plays a crucial role in preventing misuse of text generative models, which excel in mimicking human writing style recently. Latest proposed detectors usually take coarse text sequence as input and output some good results by fine-tune pretrained models with standard cross-entropy loss. However, these methods fail to consider the linguistic aspect of text (*e.g.*, coherence) and sentence-level structures. Moreover, they lack the ability to handle the low-resource problem which could often happen in practice considering the enormous amount of textual data online. In this paper, we present a **co**herence-based **co**ntrastive learning model named CoCo to detect the possible MGT under low-resource scenario. Inspired by the distinctiveness and permanence properties of linguistic feature, we represent text as a coherence graph to capture its entity consistency, which is further encoded by the pretrained model and graph neural network. To tackle the challenges of data limitations, we employ a contrastive learning framework and propose an improved contrastive loss for making full use of hard negative samples in training stage. The experiment results on two public datasets prove our approach outperforms the state-of-art methods significantly.

1 Introduction

Thriving progress in the field of text generative models (TGMs)(Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019; See et al., 2019; Lewis et al., 2020; Keskar et al., 2019; Dathathri et al., 2020; Gao et al., 2020) enables everyone to produce MGTs massively and rapidly. However, the accessibility to high quality TGMs is prone to cause misuse, such as fake news generation(Zellers et al., 2019a; Brown et al., 2020a; Yanagi et al., 2020), product review forging(Adelani et al., 2020), and spamming(Tan et al., 2012), etc. MGTs are hard to distinguish by untrained human for their human-like writing style(Ippolito et al., 2020) and the excessive amount(Grinberg et al., 2019), which calls for the study of reliable automatic MGT detector.



Points of Progress: Robots replace child camel jockeys, and more

	Human-generated	Machine-generated
Document	**S1**:Robots are taking the place of child jockeys in a major traditional Egyptian camel racing festival. **S2**:The Bedouin sport has long used children for their small size and light weight. **S3**:But the practice has come under criticism by human rights organizations that say some child jockeys are abused and malnourished. **S4**:This year, 20 of about 150 camels were controlled by robots instead of children as bans in neighboring countries and local campaigns add pressure to camel owners to change.	**S1**:At the Risk Asia conference, you could find newsmen and futurists moving back from orbit: drones, augmented reality, VR, robotics, energy, and more. **S2**:You could have even mentioned robots. **S3**:I say, what happened to science? **S4**:People got productivity and artificial intelligence out of robotics, and it was exciting. **S5**: You could find big scientists as well as TV luminaries out and back again. **S6:** These were all interesting talk that was entertaining, but as the web has progressed, you have not seen any new species or new ideas on one of these topics in decades.
Sentence Interaction		

Figure 1: Illustration of sentence-level structure difference between human-generated text and machine-generated text. Human-generated text is more coherent than machine-generated text as the sentences share more same entities even though there appears more machine-generated sentences

Previous works on MGTs detection mainly concentrate on sequence feature representation and classification(Gehrmann et al., 2019; Solaiman et al., 2019; Zellers et al., 2019a). Recent studies have shown the good performance of automated detectors in a fine-tuning fashion(Solaiman et al., 2019). Although the fine-tuning based detectors has demonstrated their effectiveness, they still suffer from two issues which limit their conversion to practical use: (1) Existing detectors treat input documents as flat sequences of tokens and use neu-

[*] All the authors contributed equally to this work.
Code and datasets are available at https://github.com/ZachYC/Coh-MGT-Detection.

ral encoder or statistical features (e.g. TF-IDF) to represent text as the dense vector for classification. These methods rely much on the token-level distribution difference of texts in each class, which ignores high-level linguistic representation of text structure like sentence interaction and text coherence. (2) Compared with the enormous number of online texts, annotated dataset for training MGT detectors is rather low-resource. Constrained by the amount of available annotated data, traditional detectors sustain frustrating accuracy and even collapse during test stage.

As shown in Fig 1, MGTs and HWTs exhibit difference in terms of coherence traced by eneity consistency. Thus, we propose an entity coherence graph to model the sentence-level structure of texts based on the thoughts of Centering Theory(Grosz and Sidner, 1986) which evaluates text coherence by entity consistency. Entity coherence graph treats entity as nodes and builds edges between entities in the same sentences and same entities among different sentences to reveal the text structure. Instead of treating text as flat sequence, coherence modeling helps to introduce distinguishable linguistic feature at input stage and provides explainable difference between MGTs and HWTs.

To alleviate the low-resource problem in the second issue, inspired by the resurgence of contrastive learning(He et al., 2020; Chen et al., 2020), we utilize proper design of sample pair and contrastive process to learn fine-grained instance-level features under low resource. However, it has been proven that the easiest negative samples are unnecessary and insufficient for model training in contrastive learning (Cai et al., 2020). To circumvent the performance degradation brought by the easy samples, we propose a novel contrastive loss with capability to reweight the effect of negative samples by difficulty score to help model concentrate more on hard samples and ignore the easy samples. Extensive experiments on multiple datasets demonstrate the effectiveness of our proposed method.

In summary, our contributions are summarized as follows:

- **Coherence Graph Construction:** We model the text coherence with entity consistency and sentence interaction while statistically proving its distinctiveness in MGTs detection, and further introduce this linguistic feature at input stage.
- **Hard Negative Mining Loss:** We propose a novel contrastive loss in which hard negative samples are paid more attention to for improving detection accuracy of challenging sample.
- **Outstanding Performance:** We achieve state-of-art performance on two MGT datasets in both low-resource and high-resource setting. Experimental results verify the effectiveness of our model.

2 Related Work

This section reviews the related works that are relevant to our work.

Machine-generated Text Detection. Machine-generated texts, also named deepfake or neural fake texts, are generated by language models and mimic human writing style, making them perplexing for humans to distinguish (Ippolito et al., 2019). Generative models like GROVER(Zellers et al., 2019b), GPT-2(Radford et al., 2019) and GPT-3(Brown et al., 2020b) has been evaluated on the MGT detection task and achieve good results. Bakhtin et al. (2019) trained an energy-based model by treating the output of TGMs as negative samples and demonstrated the generalization ability of this model on MGT detection task. Deep learning models that incorporate stylometry and external knowledge are also feasible for improving the performance of MGT detectors (Uchendu et al., 2019; Zhong et al., 2020a). Our method differs from the previous work by analyzing and modeling text coherence as distinguishable feature and emphasizing the performance improvement under low-resource scenario.

Coherence Modeling. For generative models, coherence is the critical requirement and vital target (Hovy, 1988). Previous works mainly discuss two types of coherence, local coherence (Mellish et al., 1998; Althaus et al., 2004) and global coherence (Mann, 1987). Local coherence focus on sentence-to-sentence transitions (Lapata, 2003), while global coherence tries to capture comprehensive structure (Karamanis and Manurung, 2002). Our method strives to represent both local and global coherence with inner- and inter-sentence relations between keyword nodes. Local coherence is presented as the way in which nodes relate to each other. The keyword nodes rich in semantic feature help to demonstrate how the text maintains the overall topic, which is defined as global coherence.

Contrastive Learning. Contrastive learning is a

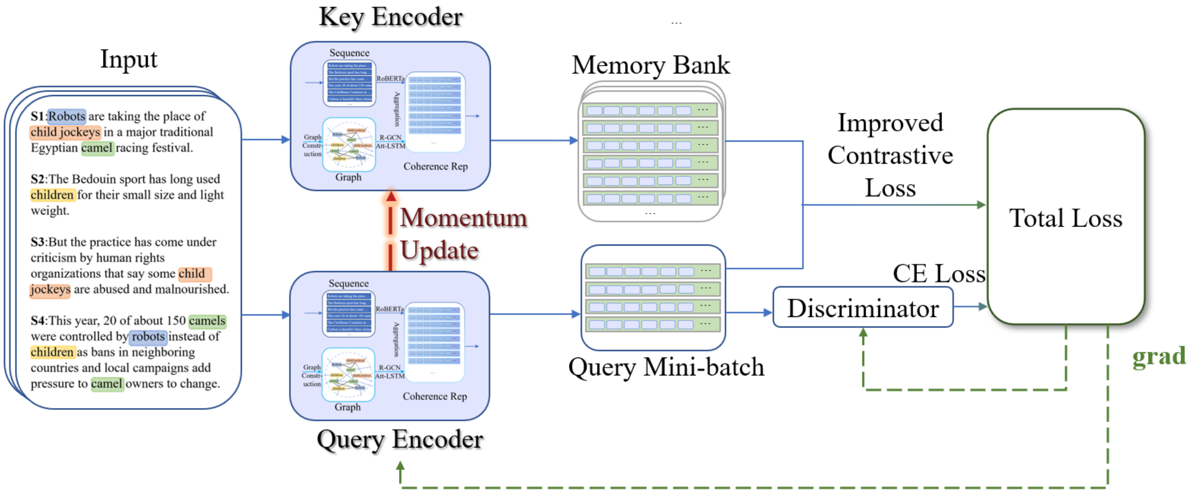

Input
S1:Robots are taking the place of child jockeys in a major traditional Egyptian camel racing festival.
S2:The Bedouin sport has long used children for their small size and light weight.
S3:But the practice has come under criticism by human rights organizations that say some child jockeys are abused and malnourished.
S4:This year, 20 of about 150 camels were controlled by robots instead of children as bans in neighboring countries and local campaigns add pressure to camel owners to change.
Key Encoder
Sequence
RoBERTa
Aggregation
Graph Constr-uction
R-GCN Att-LSTM
Graph
Coherence Rep
Momentum Update
Query Encoder
Memory Bank
Query Mini-batch
Improved Contrastive Loss
CE Loss
Discriminator
Total Loss
grad

Figure 2: Overview of CoCo. Input document is parsed as a coherence graph (4.1), the text and graph are encoded and aggregated to generate coherence-enhenced representation (4.2). After that, we employ a MoCo based contrastive learning architecture with imporved contrastive loss to make final prediction (4.3).

metric-based approach to learn discriminative representation of samples by data augmentation and predicting whether two augmented samples are from same original sample or not. In the field of NLP, contrastive learning demonstrates superb performance in learning token-level embeddings (Su et al., 2021) and sentence-level embeddings (Gao et al., 2021) for natrual language understanding. Contrastive learning also exhibits competencies in supervised setting. Gunel et al. (2020) achieves similar success with supervised contrastive loss on GLUE dataset in both full dataset and few-shot settings. With in-depth study of the mechanism of contrastive learning, the hardness of samples is proved to be crucial in the training stage. Cai et al. (2020) defined the dot product between the query and negative in normalized embedding space as hardness and figured out the easiest 95% negatives are insufficient and unnecessary. Song et al. (2022) proposed a difficulty measure function based on the distance between classes and applied curriculum learning to the sampling stage. Differently, our method pays more attention to hard negative samples for improving the detection accuracy of challenging samples.

3 Problem Formulation

This work aims to distinguish MGT from HWT with data limitation which often happens in real scenarios. The problem could be explained as a binary low-resource classification problem. Given a susceptible document d with n sentences, coherence model based on graph representation $G_c = \{\mathcal{V}, \mathcal{E}\}$ is constructed to indicate the interactions among the sentences. After encoding the coherence graph and text sequence into latent representations, a supervised contrastive learning model is designed to distinguish the text. Therefore, this detection problem could be defined simply as following.

Input: A small document set $S_d = \{d_1, d_2, \ldots, d_k\}$.

Output: With the given dataset, the model is required to make the judgement about which ones are human-written and which ones are machine-generated. A set of label S_L in return.

4 Methodology

Our workflow of CoCo mainly contains coherence graph construction, encoder, and supervised contrasive learning discriminator, and Fig. 2 illustrates its overall architecture.

4.1 Coherence Graph Construction

In this part, we illustrate how to construct coherence graph to dig out coherence structure of text by modeling sentence interaction.

According to Centering Theory(Grosz and Sidner, 1986), coherence of texts could be modeled by sentence interaction around center entities. To better reflect text structure and avoid semantic overlap, we choose entities instead of sentences as nodes. We first implement the ELMo-based NER model TagLM (Peters et al., 2017) to extract the entities from document. An inter edge is constructed between same entities in different sentences and an inner edge reflects the inclusion relationship of sentence and entity. Formally, the mathamatical form of coherence graph is defined as

$$A_{ij} = \begin{cases} 1 & label\ \langle\texttt{inner}\rangle & E_i \in s_k,\ E_j \in s_k,\ s_k \in S \\ 1 & label\ \langle\texttt{inter}\rangle & E_i = E_j\ ,\ E_i \in s_k,\ E_j \notin s_k \\ 0 & label\ \text{None} & others \end{cases}$$

4.2 Encoder Design

In this part, we introduce how to initialize node representation and graph neural network structure which is utilized to integrate coherence information into semantic representation of text by propagate and aggregate information from different granularity.

4.2.1 Node Representation Initialization

We initialize the representation of entity nodes $\mathcal{V}$ with powerful pre-trained model RoBERTa for its superior ability to encode contextual information into text representation.

Given an entity e with a span of n tokens, we utilize RoBERTa to map tokens to last layer embeddings $\boldsymbol{h}(\boldsymbol{x})$. The contextual representation of e is calculated as follows:

$$Z_e = \frac{1}{n}\sum_{i=0}^{n} \boldsymbol{h}(\boldsymbol{x})_{e_i}, \tag{1}$$

where e_i is the absolute position where the i^{th} token in e lies in the whole document.

4.2.2 Relation-aware GCN

Based on the vanilla Graph Convolutional Networks (Kipf and Welling, 2016), we propose a novel method to suit our multi-relation graph representation, which is named Relation-aware GCN. Relation-aware GCN convolute edges of each kind of relation in the graph separately. We initialize separated GCN models for different relations with different initial weights. The final representation is the sum of GCN outputs from all relations. We use two-layer GCN in the model because more layers will cause an underfitting problem under data limitation. The calculation formula is as follows. $X = [Z_{e_1}, Z_{e_2}, ..., Z_{e_n}]$ is the nodes representation input.

$$Z = \sum_{r \in R} \hat{A}_r \text{ReLU}((\hat{A}_r X W^{(0)})W^{(1)}) \tag{2}$$

$$\hat{A}_r = \tilde{D_r}^{-\frac{1}{2}} \tilde{A}_r \tilde{D_r}^{-\frac{1}{2}} \tag{3}$$

Here $\tilde{A}_r = A_r + I_N$, A_r is the adjacency matrix of the edges with relation label r. $\hat{A}_r$ is the normalized Laplacian matrix of $\tilde{A}_r$. $R = r_1, r_2, ..., r_m$ is the set of all kinds of relation labels.

Afterward, we aggregate node representation from GCN into sentence-level representation to prepare for concatenation with sequence representation from RoBERTa. The aggregation follows the below rule.

$$Z_{s_t} = \sum_{e_i \in s_t} Z_{e_i} \tag{4}$$

$$Z = [Z_{s_1}, Z_{s_2}, ..., Z_{s_n}] \tag{5}$$

4.2.3 Attention LSTM

We add a self-attention layer to the backbone of LSTM, aiming to evaluate the importance of edges in the graph representation since GCN can not deal with a weighted graph. The formula is as follows.

$$Z_{graph} = \text{softmax}(\alpha \frac{\text{norm}(Z)\text{norm}(Z)^T}{\sqrt{d_Z}}) Z * \text{LSTM}(Z) \tag{6}$$

In the formula, d_Z is the dimension of representation Z, and α is a hyper-parameter for scaling.

Finally, we get the coherence representation Z_{coh} by concatenating Z_{graph} and last layer representation $\boldsymbol{h}([\text{CLS}])$ from RoBERTa.

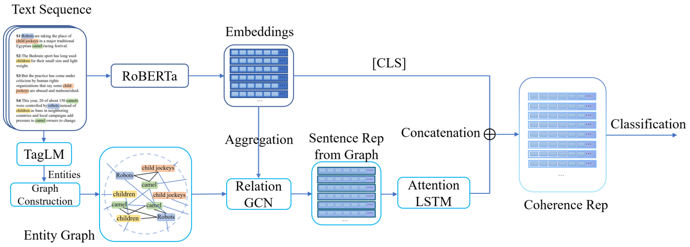


Figure 3: Encoder Architecture

4.3 Supervised Contrastive Learning

In this part, we implement a contrastive learning framework with improved contrastive loss as classifier for coherence-enhanced representation to fit the model in low-resource condition.

4.3.1 Positive/Negative Pair Definition

In supervised setting, where we have the label information, we define two samples with same label as positive pair and that with different labels as negative pair for incorporating label information into training process.

4.3.2 Model Overview

To maintain great performance under low-resource condition, we follow the architecture of MOCO(He et al., 2020) to maintain sufficient number of contrastive pairs under low-resource scenario.

The model contains two separate encoder f_k and f_q respectively with the same initialization, which are used for generating coherence-enhanced representation for given document d_k and d_q.

$$Z^k_{coh} = f_k(d_k) \tag{7}$$

$$Z^q_{coh} = f_q(d_q) \tag{8}$$

Z^k_{coh} is stroed in a dynamic memory bank with size M to store all key encoding and label information. In every training step, the newly encoded key graphs update memory bank following First In First Out(FIFO) rule to keep it updated while staying as consistent as possible. Z^q_{coh} is fed into a linear classifier to make final prediction.

$$\hat{y} = argmax(\text{softmax}(W_c Z^q_{coh} + b_c)) \tag{9}$$

where W_c and b_c are weights and bias of linear classifier.

4.3.3 Loss Function

Following the definition of positive pairs and negative pairs above, traditional supervised contrastive loss (Gunel et al., 2020) treats all positive pairs and negative pairs equally and is defined as:

$$\mathcal{L}_{\text{SCL}} = -\sum_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{M} \log \mathbf{1}_{y_i=y_j} \frac{\exp(Z^{q,i}_{coh} Z^{k,j}_{coh}/\tau)}{\sum\limits_{k=1}^{M} \exp(Z^{q,i}_{coh} Z^{k,k}_{coh}/\tau)}, \tag{10}$$

where N is the size of query set, $\mathbf{1}$ is a binary operator which denotes 1 when $y_i = y_j$ and 0 when $y_i \neq y_j$, $Z^{q,i}_{coh}$ is the i-th query encoding, and τ is a temperature hyperperameter.

With recognition that not all negatives are created equal (Cai et al., 2020), our goal is to emphasis the informative samples for helping model to differentiate difficult samples. Thus, we propose an improved contrastive loss based on Eq.10 which dynamically adjust the weight of negative pair similarity according to the hardness of negative samples. To be specific, the hard negative samples should be assigned larger weight for stimulating the model to better pull same class together and push different class away. The improved contrastive loss is defined as:

$$\mathcal{L}_{\text{ICL}} = \sum_{j=1}^{M} \mathbf{1}_{y_i=y_j} \log \frac{S_{ij}}{\sum_{p \in P(i)} S_{ip} + \sum_{n \in N(i)} rf_{in} S_{in}} \tag{11}$$

$$rf_{ij} = \beta \frac{Z^{q,i}_{coh} Z^{k,n}_{coh}}{avg(Z^{q,i}_{coh} Z^{k,1:N}_{coh})} \tag{12}$$

$$S_{ij} = exp(Z^{q,i}_{coh} Z^{k,j}_{coh}/\tau) \tag{13}$$

where $P(i)$ is the positive set in which data has the same label with q_i and $N(i)$ is the negative set in which data has different label from q_i.

Apart from instance-level learning mechanism, a linear classifier combined with cross entropy loss $\mathcal{L}_{\text{CE}}$ is employed to provide the model with class-level separation ability. $\mathcal{L}_{\text{CE}}$ is calculated by

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum\nolimits_{i=1}^{N} \mathbf{1}_{\widehat{y}_i, y_i} \log(\widehat{p}_i), \tag{14}$$

where $\mathbf{1}_{\widehat{y}_i, y_i}$ indicates whether the model makes correct prediction for i-th sample, it returns 1 when the prediction is correct and 0 otherwise, and $\widehat{p}_i$ is the predicted probability of i-th sample. The final loss $\mathcal{L}_{\text{total}}$ is a weighted average of $\mathcal{L}_{\text{ICL}}$ and $\mathcal{L}_{\text{CE}}$

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{ICL}} + (1-\alpha)\mathcal{L}_{\text{CE}}. \tag{15}$$

Here, the hyperparameter α adjusts the relative balance between instance compactness and class separability.

4.3.4 Momentum Update

The parameters of query encoder f_q and the classifier can be updated by gradient back-propagated from $\mathcal{L}_{\text{total}}$. We denote the parameters of f_q as θ_q, the parameters of f_k as θ_k, The key encoder f_q's parameters are updated by momentum update mechanism:

$$\theta_k \leftarrow \beta\theta_k + (1-\beta)\theta_q \tag{16}$$

Here, the hyperparameter β is momentum coefficient.

5 Experiments

In this section, we discuss a set of experiments to evaluate the utility and sensibility of our method CoCo, including model comparison and ablation study. The implementation about adjusting hyperparameters, model details, sampling strategy comparison, and trials on imbalanced limitation are attached in the appendix.

5.1 Experiment Settings

To ensure our CoCo is comparable, we set the experiment situation following most previous works.

5.1.1 Datasets

We evaluate our model on the following datasets:

- **GROVER Dataset** is a News-Style dataset provided by Zellers et al. (2019a), in which HWTs are collected from RealNews, a large corpus of news articles from Common Crawl, and MGTs are generated by Grover-Mega, a transformer-based news generator with parameters sized 1.5B.
- **GPT-2 Dataset** is a Webtext-style dataset provided by OpenAI[3] (Radford et al., 2022) with HWTs adopted from WebText and MGTs produced by GPT-2 XLM-1542M.

We choose $top-p$ ($p = 0.96$) as the sampling strategy for the generator based on the finding that $top-p$ leads to better text generation((Ippolito et al., 2020)). The statistic of datasets is summarized in Table 1.

Dataset		Train	Valid	Test
GROVER	HWT	5,000	2,000	8,000
	MGT	5,000	1,000	4,000
GPT-2	HWT	25,000	5,000	5,000
	MGT	25,000	5,000	5,000

Table 1: Basic Statistics of Datasets

To imitate the situation of low data-resources, we sample 10% texts from the datasets as a few-shot case, which will test models together with the complete datasets.

5.2 Methods Comparison

As machine-generated detection is an emerging task, there are no systemic method categories. From our perspective, current methods can be divided as follows.

i) Transformers Based Fine-tuning Methods. Fine-tuning is the backbone of all the detection methods, and was widely used in machine-generated tasks. Previous works purpose that vanilla fine-tuning methods based on Transformers-based models perform well. For fair compare, we choose models with approximately the same order of magnitude parameters.

- **GPT-2.**(Ippolito et al., 2020) We use the architecture of GPT-2 Small, which has 124M parameters.
- **Roberta.**(Liu et al., 2019) We follow the architecture of the cased base version of Roberta with 110M parameters.
- **XLNet.**(Yang et al., 2019) We utilize the architecture of the cased base version of XLNet, which also obtains 110M parameters.
- **GROVER.**(Zellers et al., 2019b) As GROVER mainly relies on fine-tuning to discriminate and is built based on the architecture of GPT-2, we take the base version of GROVER discriminator into account, which has 124M parameters on par with Bert-base and GPT-2 small.

Considering the conclusion that generators perform best when detecting text generated by themselves, we treat **GROVER** and **GPT-2** as baselines. We also select **Roberta** and **XLNet** as an addition for their superior performance on neural text detection reported by OpenAI(Solaiman et al., 2019) and other tasks.

ii) Contrastive Learning based Methods. We push forward our novel contrastive learning method in CoCo to suit the detection task under limitation. To show our advantage, we apply these start-of-the-art contrastive learning methods to the task with the same structure as the model's other parts.

- **CE+SCL**(Gunel et al., 2020) is the model trained with Cross-Entropy and standard supervised contrastive loss calculated within a mini-batch.
- **DualCL**(Chen et al., 2022) augment the data with classifier features and implement contrastive learning on input samples and augmented classifier samples.

We use **RoBERTa** as backbone of contrastive learning methods above.

iii) Other multi-model Methods. Only a few works have been finished to explore a multi-model

[3] https://github.com/openai/gpt-2-output-dataset

Dataset	**GOVER**				**GPT-2**			
Size	Limited Dataset (10%)		Full Dataset		Limited Dataset (10%)		Full Dataset	
Model	ACC	F1	ACC	F1	ACC	F1	ACC	F1
GPT2	64.01 ± 1.36	42.89 ± 4.13	82.74 ± 0.91	80.03 ± 1.41	85.75 ± 0.41	84.06 ± 0.70	89.13 ± 0.66	88.39 ± 0.78
XLNet	69.06 ± 3.21	51.93 ± 3.65	81.56 ± 0.79	74.93 ± 0.73	88.37 ± 0.31	87.32 ± 0.41	90.91 ± 0.91	90.27 ± 1.11
RoBERTa	77.30 ± 1.21	63.70 ± 1.86	87.72 ± 0.29	81.71 ± 0.48	92.44 ± 0.41	92.14 ± 0.45	94.02 ± 0.39	93.84 ± 0.44
DualCL*	*69.26 ± 6.34*	*53.97 ± 9.71*	*75.74 ± 8.55*	*63.88 ± 13.0*	*78.74 ± 12.1*	*69.50 ± 9.44*	*80.23 ± 11.2*	*80.46 ± 15.3*
SCL	77.77 ± 1.34	64.47 ± 2.86	87.82 ± 0.44	82.02 ± 0.57	92.87 ± 0.49	92.68 ± 0.41	94.08 ± 0.06	93.90 ± 0.09
CoCo	**78.08 ± 0.44**	**65.43 ± 1.06**	**88.26 ± 0.18**	**82.65 ± 0.36**	**92.96 ± 0.33**	**93.45 ± 0.41**	**94.28 ± 0.04**	**94.14 ± 0.04**

Table 2: Results of the Model Comparison

method to increase the detection ability of the machine. Most previous works go no further than analysis based on single Transformers-based fine-tuning Methods.

- **FAST** (Zhong et al., 2020b) combine Roberta with GCN to capture wiki-based information factual structure without significantly increasing the number of parameters of the model.

We execute our detection task in an unpaired setting, which means we provide the detector with single article, the model must classify each independently as HWT or MGT. We mainly focus on the metrics of accuracy and F1-score. Table 2 shows the performance of all the compared models and our CoCo, in which case CoCo is advanced.

As shown in Table 2, our method surpasses state-of-the-art methods in MGT detection task by at least **0.31** and **0.96** in terms of accuracy and F1-score on limited GROVER dataset. And it outstrips at least **0.09** accuracy and **0.77** F1-score on limited GPT-2 dataset. On full dataset, CoCo outperforms other models by at least **0.44** and **0.63** in metrics of accuracy and F1-score on GROVER. CoCo also towers above at least **0.20** accuracy and **0.24** F1-score on GPT-2 dataset.

Moreover, our model also excels the test result reported by GROVER and FAST. The accuracy score of GROVER-Base with about the same size of CoCo 80.0 on its dataset. And FAST reports its accuracy on GROVER is 84.90 and on GPT-2 is 93.10. Our CoCo outperforms both of them.

It indicates that CoCo has significant power of mining distinguishable features and suits for both low-resource and high-resource environment. Moreover, it should be noticed that CoCo gets less affected by randomness, which illustrates that the coherence graph we construct is a robust feature that helps stabilize the model performance. Additionally, we find that the randomness intensely impacts DualCL, and the model is very prone to failure, leading to a large standard deviation in Table 2. It is because DualCL is mainly focused on evaluating short texts which contains only few words (around 10 words) and is not suitable for long-term evaluation.

5.3 Ablation Study

To illustrate the necessity of every part of our CoCo and their enhancement to the model performance, we carry out five groups of ablation experiments as follows. By ablation and derivation, trials embody CoCo's structure reliability and extension potentials. Table 3 shows the results of the ablation study. A detailed explanation of our ablation settings is as follows. The dataset here is 10% GROVER.

Model	**ACC**	**F1**
w/o Graph & CL	76.97	64.28
w/ Sent. Graph w/o CL	77.33	63.79
w/ Graph w/o CL & LSTM	77.77	64.63
w/ Graph w/o CL	77.87	64.71
w/ Graph & SCL	78.27	66.09
CoCo	78.43	66.84

Table 3: Results of Ablation Study

w/o Graph & CL. removes graph information and encodes only by RoBERTa parts. Moreover, the model removes contrastive learning and uses CE loss.

w/ Sent. Graph w/o CL. treats sentences as nodes instead of entities and removes contrastive learning and uses CE loss.

w/ Graph w/o CL & LSTM. removes contrastive loss and replace LSTM in encoder with average pooling operation.

w/ Graph w/o CL . uses LSTM for document-level aggregation and the rest is the same as w/ Graph w/o CL & LSTM.

w Graph & SCL . replaces the improved contrastive loss with traditional supervised contrastive loss.

As shown in Table 3, we can see that CoCo outperforms baseline models even after ablation. Though the performance of ablated models slightly drops compared to the completed CoCo. This result proves the feasibility and sensibility of our design of CoCo architecture and shows its potential to extend.

6 Discussion

6.1 Static Geometric Analysis on Coherence Graph

We have witnessed performance enhancement by applying the graph-based coherence model to the detection model, but how does the coherence graph help detection? In this subsection, we apply static geometric features analysis to coherence graph we construct to evaluate the distinguishable difference between HWTs and MGTs with explaination. In the following discussion, we take the dataset of GROVER into the analysis. Some basic metrics of data and the corresponding graph are shown in Table 4.

Metric	**HWT**	**MGT**
Sample Num.	4994	4991
Avg. Num. of Token	463.2	456.0
Avg. Num. of Vertex	43.60	32.37
Avg. Num. of Edge	107.4	65.44

Table 4: Basic Metrics of Texts and Corresponding Graphs

Though HWTs and MGTs have approximately the same number of tokens in every text, coherence graph for HWTs has larger scale than MGTs' with **34.7%** more vertexes and **64.1%** more edges, which shows that HWTs have more complex semantic relation structures than MGTs.

6.1.1 Degree Distribution

Semantically, degree of coherence graph measures the co-occurrence and TF-IDF feature of keywords. Moreover, degree distribution shows global coherence because high-degree nodes devote to the main topic and low-degree nodes are the extension.

Metric	**Avg. Degree**
HWT	2.980
MGT	2.591

Table 5: Avg of Degree (Whole Dataset)

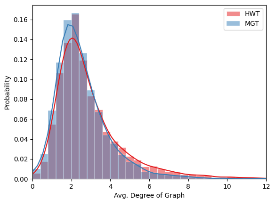


Figure 4: Distribution of Avg. Degree of Graphs

As shown in Table 5, the degree of the graph representation of HWTs is **15.0%** larger than MGTs, which shows disparities of MGTs to form coherent interaction between sentences. Fig. 4 measures the distribution of each graph's average nodes' degree, showing that the distribution of HWTs has a longer tail than MGTs.

Furthermore, we analyze the distinguishability of degree features when impacted by other factors. One most considerable influences is the style and genre of different provenance. We chose around 60 articles from The Sun[4] and Boston[5]. Then we use GROVER to mimic their style to generate similar topic news. Fig. 5 shows the degree distribution of HWTs and MGTs of both provenances.

We use Jensen–Shannon divergence to evaluate the similarity of the degree distribution. The JS-divergence of MGTs mimicking TheSun and Boston is **0.029**, while the JS-divergence of MGTs and HWTs in Boston is **0.050**, in TheSun is **0.061**. The apparent gap shows that degree distribution can robustly detect MGTs and HWTs when impacted by provenance differences.

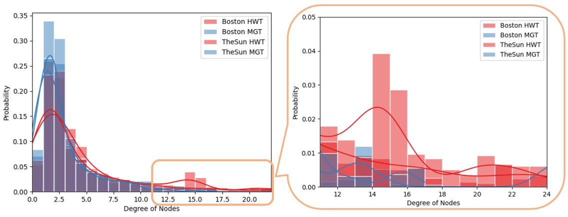


Figure 5: Distribution of Degree with different provenance

[4] https://www.thesun.co.uk/

[5] https://www.boston.com/

6.1.2 Aggregation

Aggregation is a shared metric for complex networks and linguistics, depicting how closely the whole is organized around its core. We propose two metrics to evaluate the aggregation of graph-based text representation in our coherence model, the size of the largest connected subgraph and the clustering coefficient.

In our representation, not all sentences have entities related to others. Hence the graph is an unconnected one. The average number of nodes in subgraphs of MGTs is **4.49** and of HWTs is **4.84**. We propose that the size of the largest connected subgraph shows the contents which are closely organized around the topic. Moreover, the size of graphs may be an unfair factor, so we use the portion of nodes in the largest connected subgraph to reflect its size. The average portion in HWTs is **0.6725** and in MGTs is **0.6458**. Fig. 6 shows the distribution of the portion of graphs, and HWTs distribute more high-portion ones than MGTs.

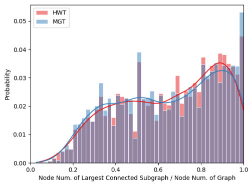


Figure 6: Portion of the Largest Connected Subgraph

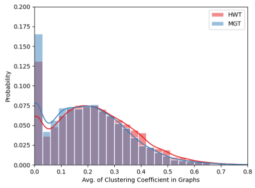


Figure 7: Distribution of Clustering Coefficient

The clustering coefficient represents how nodes tend to cluster. For the entities of texts, clustering evaluates how the author narrates around the central theme. The larger the clustering coefficient is, the tighter the semantic structure is. The average cluster coefficient of the graphs of HWTs is **0.2213** and of MGTs is **0.1983**, HWTs is **11.6%** better than MGTs. Fig. 7 shows the distribution.

6.1.3 Core & Degeneracy

The degeneracy of a graph is a measure of how sparse it is, and the k-core is the subgraph corresponding to its significance in the graph. We propose that, in our graph representation, the degeneracy process of graphs equals summarizing texts semantically. The maximum of core-number shows the complexity of hierarchical structure in texts. Furthermore, the distribution of the core-number reflects the overall sparse and is a graph-perspective N-gram module. Based on experiments, the average core-number of HWTs is **5.772** while MGTs with **4.458**. HWTs are **29.5%** ahead. Fig. 8 is the distribution of the core-number.

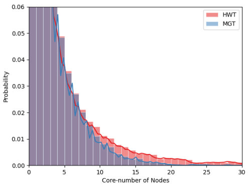


Figure 8: Core-number of Nodes in Graphs

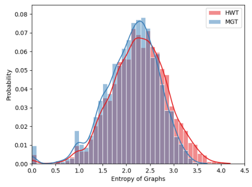


Figure 9: Structure Entropy of Graphs

6.1.4 Entropy

Entropy is a scientific concept to measure a state of disorder, randomness, or uncertainty. The well-known Shannon entropy is the core of the information theory, measuring the self-information content. For the graph data, network structure entropy defined as the following can examine the information amount of the graph structure.

$$E = -\sum_{i=1}^{N} I_i \cdot \ln I_i = -\sum_{i=1}^{N} \frac{k_i}{\sum_{j=1}^{N} k_j} \cdot \ln\left(\frac{k_i}{\sum_{j=1}^{N} k_j}\right) \tag{17}$$

Global coherence, from our perspective, equals refining more information inside the semantic structure of the whole text, which matches to structure entropy of our graph representation. From our experiments, the structure entropy of HWTs (2.263) is **6.80%** larger than MGTs (2.119), which means HWTs obtain more structured information because their semantic information is globally organized. We show the network structure entropy distribution in Fig. 9.

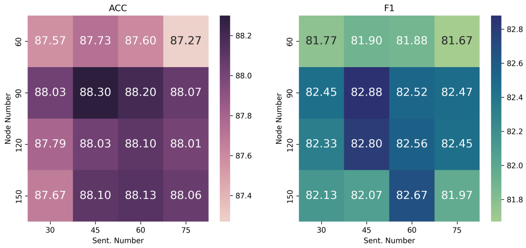


Figure 10: Performance of Model with Different Graph Parameters

6.2 Effect of Graph Parameters

We further investigate the effect of max node number and max sentence number on model performance. The result is shown in Fig 10. We select max node number from $\{60, 90, 120, 150\}$ and max sentence number from $\{30, 45, 60, 75\}$. The detector performs best when max node number is 90 and max sentence number is 45. The experiment results prove that the large node and sentence number are not necessary for the improvement of detection accuracy. We infer that even though setting large node and sentence number includes more entity information, excessive nodes bring noise to the model and impair the distinguishability of coherence feature.

7 Conclusion

In this paper, we propose CoCo, a coherence-enhanced contrastive learning model for MGT detection. We implement a MoCo-based contrastive learning framework to improve model performance in low-resource setting. The encoder in CoCo is composed of sequence representation and coherence representation. We construct a novel coherence graph from document, which is further utilized to learn coherence representation by relation-aware GCN and attention LSTM to aggregate graph information into document-level coherence representation. Coherence-enhanced representation of document is composed of sequence representation extracted by RoBERTa and learned coherence representation. To alleviate the effect of unnecessary easy sample, we propose an improved contrastive learning loss to force model pay more attention to hard negative samples. We evaluate our method on MGT dataset generated by GROVER and GPT-2, respectively, in both low-resource and high-resource setting. CoCo outperforms Transformer-based methods and contrastive-learning-based methods on both datasets and both settings. The static geometric analysis on coherence graph validates the difference of MGTs and HWTs from the view of coherence graph and points out the potential defects of popular TGMs' output.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In *International Conference on Advanced Information Networking and Applications*, pages 1341–1354. Springer.

Ernst Althaus, Nikiforos Karamanis, and Alexander Koller. 2004. Computing locally coherent discourses. *Untitled Event*.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *arXiv: Learning*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Thomas Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Samuel McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *neural information processing systems*.

Tiffany Tianhui Cai, Jonathan Frankle, David J Schwab, and Ari S Morcos. 2020. Are all negatives created equal in contrastive instance discrimination? *arXiv preprint arXiv:2010.06682*.

Qianben Chen, Richong Zhang, Yaowei Zheng, and Yongyi Mao. 2022. Dual contrastive learning: Text classification via label-aware data augmentation. *arXiv preprint arXiv:2201.08702*.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. *ArXiv*, abs/1912.02164.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *north american chapter of the association for computational linguistics*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *meeting of the association for computational linguistics*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116.

Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. 2019. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378.

Barbara J Grosz and Candace L Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pretrained language model fine-tuning. *arXiv preprint arXiv:2011.01403*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Eduard Hovy. 1988. Planning coherent multisentential text. *meeting of the association for computational linguistics*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *meeting of the association for computational linguistics*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822.

Nikiforos Karamanis and Hisar Maruli Manurung. 2002. Stochastic text structuring using the principle of continuity. *international conference on natural language generation*.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858.

Thomas Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv: Learning*.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. *meeting of the association for computational linguistics*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Michael Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv: Computation and Language*.

William C. Mann. 1987. Rhetorical structure theory: A theory of text organization.

Chris Mellish, Alistair Knott, Jon Oberlander, and Mick O'Donnell. 1998. Experiments using stochastic search for text planning. *international conference on natural language generation*.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2022. Language models are unsupervised multitask learners.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abigail See, Aneesh S. Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers. *conference on computational natural language learning*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Xiaohui Song, Longtao Huang, Hui Xue, and Songlin Hu. 2022. Supervised prototypical contrastive learning for emotion recognition in conversation. *arXiv preprint arXiv:2210.08713*.

Yixuan Su, Fangyu Liu, Zaiqiao Meng, Tian Lan, Lei Shu, Ehsan Shareghi, and Nigel Collier. 2021. Tacl: Improving bert pre-training with token-aware contrastive learning. *arXiv preprint arXiv:2111.04198*.

Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang, and Yihong Zhao. 2012. Spammer behavior analysis and detection in user generated content on social networks. In *2012 IEEE 32nd International Conference on Distributed Computing Systems*, pages 305–314. IEEE.

Adaku Uchendu, Jeffrey Cao, Qiaozhi Wang, Bo Luo, and Dongwon Lee. 2019. Characterizing man-made vs. machine-made chatbot dialogs. In *TTO*.

Yuta Yanagi, Ryohei Orihara, Yuichi Sei, Yasuyuki Tahara, and Akihiko Ohsuga. 2020. Fake news detection with generated comments for news articles. In *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)*, pages 85–90. IEEE.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *neural information processing systems*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019a. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019b. Defending against neural fake news. *neural information processing systems*.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020a. Neural deepfake detection with factual structure of text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2461–2470.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020b. Neural deepfake detection with factual structure of text. *empirical methods in natural language processing*.